



QUICK GROCERY STORE



Submitted to the **Bangalore North University** in partial
fulfillment of the requirement for the award of the degree in
“Bachelor of Computer Applications”

Submitted by

Aditya Venkat Ganesh P

R1819203

Bronson Takhellambam

R1819213

Under the Guidance of

Ms. Anjana Murthy

Asst. Professor

Declaration

I hereby declare that the project report entitled "**Quick Grocery Store**" was prepared by me during the year **2020-2021** and was submitted to the **Bangalore North University**, in partial fulfillment of the requirements for the award of the Degree in **Bachelor of Computer Applications**. I also declare that this project report is original and genuine and it has not been submitted to any other University for the award of any degree, diploma or other similar titles or purposes.

Date:

Place: New Horizon College (Marathalli)

Signature of the Candidate

Aditya Venkat Ganesh P
[R1819203]

Certificate

This is to certify that the project report entitled "**Quick Grocery Store**" Submitted by **Aditya Venkat Ganesh P (R1819203)**. This report is an outcome of genuine project work and has been submitted in partial fulfillment for the award of the Degree of **Bachelor of Computer Applications**, awarded by **Bangalore North University**, during the academic year **2020-21**.

Project Guide:

Prof. ANJANA MURTHY

Head of the Department

Dr. NAGARAJU KILARI

Signature of the Examiners:

1. _____

2. _____

Acknowledgement

We express our deep sincere thanks to **Dr. Mohan Maghnani, Chairman of New Horizon Educational Institutions** for providing the platform and infrastructure to do this project. This project would not have been possible without the help and co-operation of many. At the outset, I wish to record my profound gratitude to our principal **Dr. R. Bodhisatvan**

We are extremely thankful to **Dr Nagaraju K**, Head-Computer science for his unending support and encouragement during the development of this project.

We would like to acknowledge the interest and the support extended by our project guide **Ms. Anjana Murthy**, Asst. Professor, Department of Computer Science, to make this project implementation successful.

Above all we would thank God for the blessings and my parents also for their valuable suggestion and support in my project report

Finally, we extend my deep sense of gratitude to all those who made this project come alive and encouraged and guided me from the start to finish.

Index

| Chapters | Contents | Page No |
|--------------------|--------------------------------------|----------------|
| Chapter - 1 | Introduction of the project | |
| | 1.1 Abstract | 2 |
| | 1.2 Problem Statement | 2 |
| | 1.3 Objectivies | 3 |
| | 1.4 Functionalities | 3 |
| | 1.5 Scope | 4 |
| | 1.6 Modules | 4 |
| Chapter - 2 | System Analysis | |
| | 2.1 Introduction to System Analysis | 6 |
| | 2.1.1 Existing System | 6 |
| | 2.1.2 Proposed System | 8 |
| | 2.2 Requirements Specifications | 8 |
| | 2.3 Hardware Requirements | 9 |
| | 2.4 Software Requirements | 9 |
| | 2.5 Introduction to MS Visual Studio | 10 |
| | 2.6 Introduction to My SQL | 11 |
| Chapter - 3 | System Design | |
| | 3.1 ER Diagram | 12 |
| | 3.2 Table Structure | 15 |
| | 3.3 Form Design | 15 |
| | 3.4 Feasibility Study | 18 |
| | 3.5 Project Planning | 24 |
| | 3.6 Use Cases | 27 |
| Chapter - 4 | Coding & Development | 34 |
| | 4.1 Source Code | 45 |

| Chapters | Contents | Page No |
|--------------------|--|----------------|
| Chapter - 5 | Testing & Implementation | |
| | 5.1 Introduction to Testing | 58 |
| | 5.2 Types of Testing | 58 |
| Chapter - 6 | Conclusion & Future Enhancement | |
| | 6.1 Conclusion | 64 |
| | 6.2 Future Enhancement | 65 |
| | 6.3 Limitations | 65 |
| Chapter - 7 | Bibliography | 66 |

Introduction



Online shopping is a form of electronic commerce which allows consumers to directly buy goods or services from a seller over the Internet using a web browser or a mobile app. Consumers find a product of interest by visiting the website of the retailer directly or by searching among alternative vendors using a shopping search engine, which displays the same product's availability and pricing at different e-retailers.

Online stores usually enable shoppers to use "search" features to find specific models, brands or items. Online customers must have access to the Internet and a valid method of payment in order to complete a transaction, such as a credit card, an Interac-enabled debit card, or a service such as PayPal. For physical products (e.g., paperback books or clothes), the e-tailer ships the products to the customer; for digital products, such as digital audio files of songs or software, the e-tailer usually sends the file to the customer over the Internet.

The largest of these online retailing corporations are Alibaba, Amazon.com, and eBay.

Abstract

1.1 Abstract

An online shopping system that permits a customer to submit online orders for items and/or services from a store that serves both walk-in customers and online customers. The online shopping system presents an online display of an order cut offtime and an associated delivery window for items selected by the customer. The system accepts the customer's submission of a purchase order for the item in response to a time of submission being before the order cut offtime. The online shopping system does not settle with a credit supplier of the customer until the item selected by the customer is picked from the inventory but before it is delivered. Therefore, the customer can go online and make changes to the order. In addition, available service windows are presented to the customer as a function of customer selected order and service types and further, the order picking is assigned in accordance with a picker's preference. When ordering goods, many shopping systems provide a virtual shopping cart for holding items selected for purchase.

Successive items selected for purchase are placed into the virtual shopping cart until a customer completed their shopping trip. Virtual shopping carts may be examined at any time, and their contents can be edited or deleted at the option of the customer. Once the customer decides to submit a purchase order, the customer may print the contents of the virtual shopping basket in order to obtain a hard copy of the transaction.

Problem Statement

1.2 Problem Statement

Online grocers face number of challenges. The major challenge is lack of handy experience in consumer demands. Online market has developed its space in virtual world but is this market worth for all kinds of products specially the perishable grocery products. Thus, there is need to study consumer perception towards the online grocery market. The marketers need to analyze and understand what their consumer actually want and what they perceive about their online service.

Objectives

1.3 Objectives

1. Reduce management costs.
2. Developing business relations.
3. Providing a unique customer experience.
4. Increasing the number of loyal customers.
5. Boosting the efficiency of services.
6. Developing relevant target.
7. Making a responsive e-commerce website.
8. Increasing sales.

Functionalities

1.4 Functionalities

1. Cart sharing, shopping cart is considered to be one of the major yet underrated aspects of e-commerce websites.
2. Product return.
3. Multiple payment options.
4. Reviews and ratings.
5. Referrals and rewards.
6. Easy registration.
7. Delivery scheduler.
8. Quick reorder or repeat order.

Scope

1.5 Scope

The growth of e-Commerce has been one of the more positive outcomes of the recent COVID-19 crisis. This is what David Bishop (the partner at Brick Meets Click) said about the e-commerce boom from late March to early April, “It's the equivalent of five years' growth within a matter of the month.”

Now, experts are saying that it's time for grocers to improve their eCommerce stores. They are supposed to add all the necessary features and enhance the performance of their online store and mobile app. We all evolved from living in a world with no Internet to living in a world that cannot be imagined without Internet. Yes, we exist in the era of digitisation. From our daily needs to our daily activities we are surrounded by online apps, which have now become an integral part of our lives. This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24*7 and a home delivery system which can make customers happy.

If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won't be losing any more customers to the trending online shops such as flipcart or e-bay. Since the application is available in the smartphone it is easily accessible and always available.

Modules

1.6 Modules:

The system after careful analysis has been identified to be presented with the following modules and roles.

The modules involved are:

- 1). Administrator
- 2). Users

Administrator Modules:

- 1. Addcategory Module:** This is used to add categories into the website.
- 2. Additem Module:** This is used to add items/products into the website.
- 3. Addquantity Module:** This is used to add quantities to the products.
- 4. Feedback Module:** This is used to examine feedbacks from the users.
- 5. Vieworder Module:** This is used to view the orders placed by the users.
- 6. ViewPayment Module:** This is used to view the payments made by the users

User Modules:

- 1. Checkout Module:** This module manages the products to be payed and processed.
- 2. MyAccount Module:** This module contains the profile details of the user.
- 3. MyCart Module:** This module contains the products the user wishes to purchase.
- 4. MyOrder Module:** This module contains the history of the orders placed by the user.
- 5. MyPayment Module:** This module contains the various modes of payments to purchase the items.
- 6. Search Module:** This module searches the items according to the respective categories.
- 7. View Module:** This module gives the detailed information of the items with images of the products.
- 8. Feedback Module:** This Module allows the user to enter his feedbacks/opinions or experiences faced while using the website.

System Analysis

2.1 Introduction to System Analysis

A **software requirements specification** (SRS) is a description of a software system to be developed. It lays out functional and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide. Software requirements establishes the basis for an agreement between customers and contractors or suppliers (in market-driven projects, these roles may be played by the marketing and development divisions) on what the software product is to do as well as what it is not expected to do. Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. It should also provide a realistic basis for estimating product costs, risks, and schedules. Used appropriately, software requirements specifications can help prevent software project failure.

System analysis is a process of gathering and interpreting facts, diagnosing problems and the information about the Railway Management System to recommend improvements on the system. It is a problem solving activity that requires intensive communication between the system users and system developers.

2.1.1 Existing System

- A Grocery Store is a retail store that primarily sells products/foods.
- A Grocer is a bulk seller of foods.
- Grocery Stores often offers non-perishable foods that is packed in cans, bottles and boxes, with some also having fresh produce, butchers, delis and bakeries.
- As pollution around the world has increased, buying food on the roadside is not advisable. Whether in charge of a small individually owned grocery store or one that is part of larger chain maintaining a grocery store successfully involves considerable responsibility.
- Grocery Store managers must ensure that the store runs smoothly that items are priced comparatively and that customers are satisfied.

Data Dictionary:

This is normally represented as the data about data. It is also termed as metadata sometimes which gives the data stored in the database. It defines each data term encountered during the analysis and design of a new system. Data elements can describe files or the processes.

Following are some major symbols used in the data dictionary:

1. = equivalent to
2. + and
3. [] either/or
4. () optional entry

Following are the rules, which defines the construction of data dictionary entries:

1. Words should be defined to understand for what they need and not the variable need by which they may be described in the program.
2. Each word must be unique. We cannot have two definition of the same client.

Active Data Dictionary:

Active data dictionary is built into majority of database management systems (DMBS). It is accessible to users with a set of system tables or views and present information about tables, columns, data types, scripts and other objects in the database. Every database schema update (using DDL scripts, such as ALTER TABLE) is automatically reflected in active data dictionary, so it does not require any maintenance from the database admin.

Passive Data Dictionary:

Passive data dictionary is separate from the database and all changes in database structure need to be applied in the passive data dictionary manually, or with dedicated software.

Passive data dictionary can take different forms:

1. A document or spreadsheet

2. Tools

- Data Catalogs
- Data integration/ETL metadata repositories
- Data modeling tools

3. Custom implementations

2.1.2 Proposed System

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features:

- Understanding the key concepts involved in effective grocery store management is imperative for any manager dedicated to the success of his store.
- Particularly if he manages the small grocery store inventory requires more than simply keeping enough of every item in stock.
- This is a internet based application that is accessed through web.
- This system can be used to search for all grocery and fresh vegetables which are available in the market.

2.2 Requirements Specification

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. The Software Requirement Specification is documented in such a way that it breaks the deliverable into smaller components.

If your development team do not understand that there are certain constraints on their work, as for example the code must be tightly written so that it will compile and run quickly, then you will run into problems later on when the code might deliver the functionality required, but no one will ever see it because it takes so long to load.

Hardware Requirements

| | |
|---------------------------|---------------------------|
| Processor | Intel Pentium IV 2.66 GHz |
| Main Memory | 512 MB |
| Hard Disk Capacity | 160 GB |
| Monitor | VGA with Resolution |
| Keyboard | 122 Keys |

Software Requirements

| | |
|-------------------------|-------------------------|
| Operating System | Windows 10 |
| Front End | Microsoft Visual Studio |
| Back End | My SQL |
| Server | ASP.NET |

2.5 Introduction to Microsoft Visual Studio:

Visual Studio is an Integrated Development Environment(IDE) developed by Microsoft to develop GUI(Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB(Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS.

Evolution of Visual Studio: The first version of VS(Visual Studio) was released in 1997, named as Visual Studio 97 having version number 5.0. The latest version of Visual Studio is 15.0 which was released on March 7, 2017. It is also termed as Visual Studio 2017. The supported .Net Framework Versions in latest Visual Studio is 3.5 to 4.7. Java was supported in old versions of Visual Studio but in the latest version doesn't provide any support for Java language.

Visual Studio Editions;

There are 3 editions of Microsoft Visual Studio as follows:

1. Community: It is a free version which is announced in 2014. All other editions are paid. This contains the features similar to Professional edition. Using this edition, any individual developer can develop their own free or paid apps like .Net applications, Web applications and many more. In an enterprise organization, this edition has some limitations. For example, if your organization have more than 250 PCs and having annual revenue greater than \$1 Million(US Dollars) then you are not permitted to use this edition.

2. Professional: It is the commercial edition of Visual Studio. It comes in Visual Studio 2010 and later versions. It provides the support for XML and XSLT editing and includes the tool like Server Explorer and integration with Microsoft SQL Server. Microsoft provides a free trial of this edition and after the trial period, the user has to pay to continue using it. Its main purpose is to provide Flexibility(Professional developer tools for building any application type), Productivity(Powerful features such as CodeLens improve your team's productivity), Collaboration(Agile project planning tools, charts, etc.) and Subscriber benefits like Microsoft software, plus Azure, Pluralsight, etc.

3. Enterprise: It is an integrated, end to end solution for teams of any size with the demanding quality and scale needs. Microsoft provides a 90-days free trial of this edition and after the trial period, the user has to pay to continue using it. The main benefit of this edition is that it is highly scalable and deliver high-quality software.

2.6 Introduction to MySQL

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons –

1. MySQL is released under an open-source license. So you have nothing to pay to use it.
2. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
3. MySQL uses a standard form of the well-known SQL data language.
4. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
5. MySQL works very quickly and works well even with large data sets.
6. MySQL is very friendly to PHP, the most appreciated language for web development.

RDBMS Terminology:

Database – A database is a collection of tables, with related data.

Table – A table is a matrix with data. A table in a database looks like a simple spreadsheet.

Column – One column (data element) contains data of one and the same kind, for example the column postcode.

Row – A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.

Redundancy – Storing data twice, redundantly to make the system faster.

Primary Key – A primary key is unique. A key value can not occur twice in one table. With a key, you can only find one row.

Foreign Key – A foreign key is the linking pin between two tables.

Compound Key – A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.

System Design

3.1 Entity Relationship Diagram:

The E-R Model is a popular high level conceptual data model. This model and its variations are frequently used for the conceptual design of database application and many database design tools employ its concept.

A database that confirms to an E-R diagram can be represented by a collection of tables in the relational system. The mapping of E-R diagram to the entities are:

1) Attributes

2) Relations

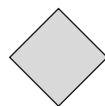
- > Many-to-Many
- > Many-to-One
- > One-to-Many
- > One-to-One

The entities and their relationships between them are shown using the following conventions:

> An entity is shown in rectangle.



> A diamond represent the relationship among number of entities.

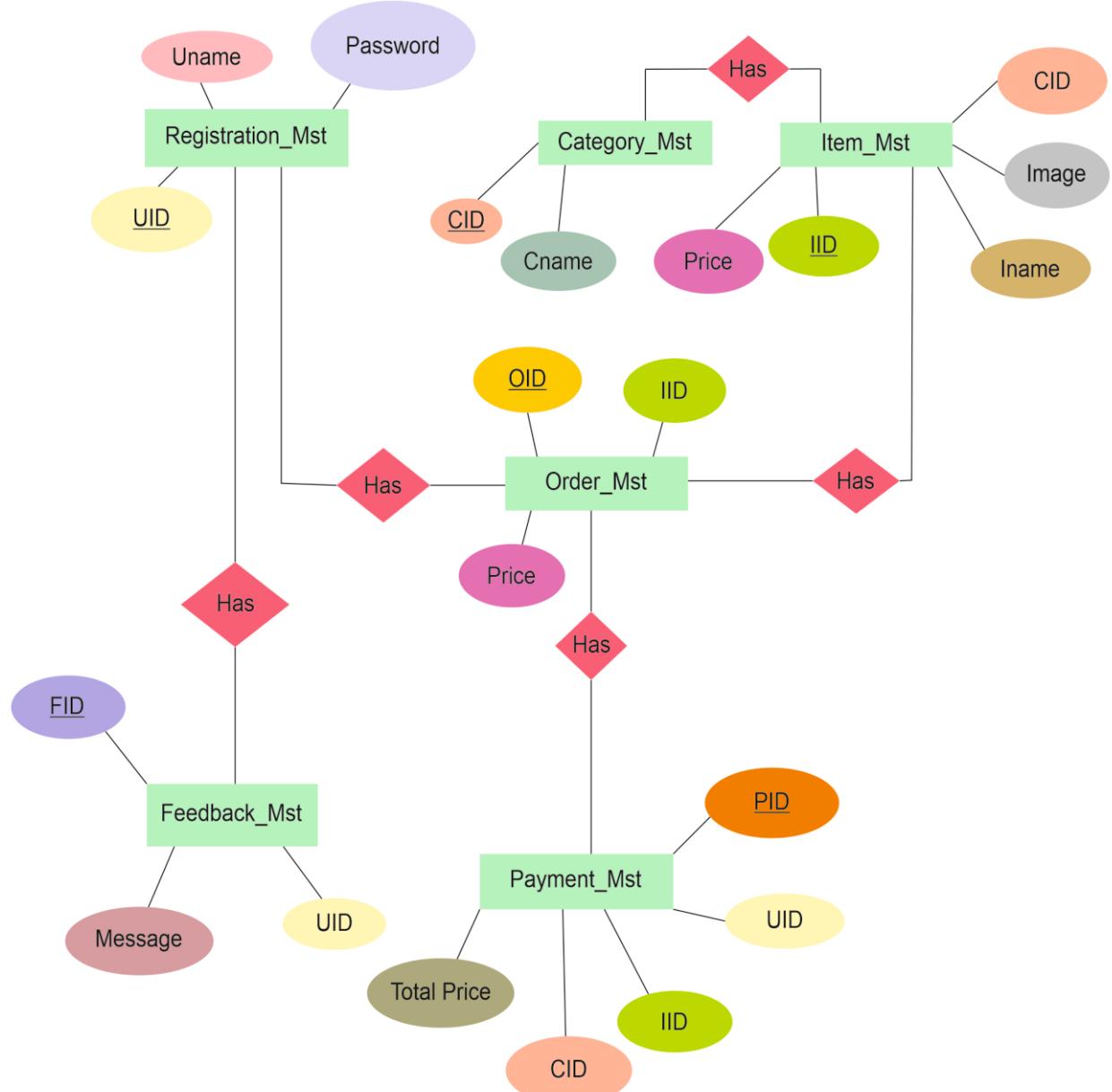


> Diamond, oval and relationships are labelled.

> A data model is a mechanism that provides this abstraction for database application.

- > Data modeling is used for representing entities and their relationship in the database.
- > Entities are the basic units used in modeling database entities can have concrete existence or constitute ideas or concepts.
- > A key is a single attribute or combination of 2 or more attributes of an entity set is used to identify one or more instances of the set.

ER Diagram



3.2 Table Structure

1. Login Tables

| SL No. | Field Name | Field Type | Constraints | Description |
|---------------|-------------------|-------------------|--------------------|---------------------|
| 1. | ID | Text | Primary Key | Contains Serial no. |
| 2. | Username | Text | NOT NULL | Contains Username |
| 3. | Pass | Text | NOT NULL | Contains Password |
| 4. | Name | Text | NOT NULL | Contains Name |
| 5. | Add | Text | NOT NULL | Contains Address |

2. Order Tables

| SL No. | Field Name | Field Type | Constraints | Description |
|---------------|-------------------|-------------------|--------------------|---------------------|
| 1. | ID | Text | Primary Key | Contains Serial no. |
| 2. | o_id | Text | NOT NULL | Contains Order_id |
| 3. | item | Text | NOT NULL | Contains the item |
| 4. | price | Text | NOT NULL | Contains the price |
| 5. | total | Text | NOT NULL | Contains the total |
| 6. | user | Text | NOT NULL | Contains username |
| 7. | status | Text | NOT NULL | Contains the Status |

3. Cart Tables

| SL No. | Field Name | Field Type | Constraints | Description |
|--------|------------|------------|-------------|---------------------|
| 1. | ID | Text | Primary Key | Contains Serial no. |
| 2. | item | Text | NOT NULL | Contains itemname |
| 3. | price | Text | NOT NULL | Contains Price |
| 4. | total | Text | NOT NULL | Contains total |
| 5. | status | Text | NOT NULL | Contains the status |

Features of the project Quick Grocery Store:

- 1) Product and item based.
- 2) Simple status and resolutions.
- 3) It contains better inventory capacity.
- 4) Accuracy product details.
- 5) Easy and fast payment of items.
- 6) Well designed reports.
- 7) Becomes fast & efficient.
- 8) Easy to update and enter the information.

Software Requirement Specification:

The software requirement specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description, a detailed design constraints, appropriate version, validation criteria, and other data pertinent to the requirements.

The proposed system has the following requirements:

- 1) System needs to store information about new entry of ticket.
- 2) System needs to help the internal staff to keep information of ticket.
- 3) System needs to maintain the quantity record.
- 4) System needs to keep the record of booking.
- 5) System also needs to search area.
- 6) It also needs a security system to prevent data loss.

Identification of need:

The old manual system was suffering from a series of drawbacks. Since whole of the system was to be maintained with hands the process of keeping, maintaining and retrieving the information was very tedious and lengthy. The records were never used to be in a systematic order.

There would always be unnecessary consumption of time while entering and retrieving the records. It was also difficult to locate for the errors. Once the record was entered it was difficult to update these records.

The Following points must be well considered:

- > Documents and reports that must be provided by the new system.
- > Details of the information needed for each document and report.
- > The required frequency and distribution for each document.
- > Probable sources of information for each document and report.
- > With the implementation of computerized system, the task of keeping records in an organized manner will be solved.
- > The greatest of all is the retrieval of information, which will be at the click of the mouse. So the proposed system helps in saving the time in different operations and making the information flow easy giving valuable reports.

3.4 Feasibility Study

After doing the project Railway Management System, study and analyzing all the existing or required functionalities of the system, the next task is to do the feasibility study for the project. All projects are feasible - given unlimited resources and infinite time.

Feasibility Study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

A. Economical Feasibility

This is a very important aspect to be considered while developing a project. We decided the technology based on the minimum possible cost factor.

- 1) All hardware and software cost has to be done by the organization.
- 2) Overall we have estimated that the benefits the organization is going to receive from the proposed system will surely overcome the initial costs and later on running costs for the system.

B. Technical Feasibility

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionality to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of frontend and backend platforms.

C. Operational Feasibility

No doubt the proposed system is fully GUI based that is very user friendly and all inputs to be taken all self-explanatory even to a layman. Besides, a proper training has been conducted to let know the essence of the system to the users so that they feel comfortable with the new system. As far our study is concerned the clients are comfortable and happy as the system has cut down their loads and doing.

System design of Quick Grocery Store:

In this phase, a logical system is built which fulfils the given requirements. Design phase of software development deals with transforming the client's requirements into a logically working system. Normally, design is performed in the following two steps:

1. Primary Design Phase

In this phase, the system is designed at block level. The blocks are created on the basis of analysis done in the problem identification phase. Different blocks are created for different functions emphasis is put on minimizing the information flow between the blocks. Thus, all activities which require more interaction are kept in one block.

2. Secondary Design Phase

In the secondary design phase the detailed design of every block is performed.

The General tasks involved in the design process are:

- 1) Design various blocks for overall system.
- 2) Design smaller, compact and workable modules.
- 3) Design various database structures.
- 4) Specify the details of programs to achieve desired functionality.
- 5) Design the form of inputs, and outputs of the system.
- 6) Perform documentation of the design.
- 7) System reviews.

User Interface Design

User interface design is concerned with the dialogue between a user and the computer. It is concerned with everything from starting the system or logging into the system to the eventually presentation of desired inputs and outputs. The overall flow of the screens and messages is called a dialogue.

The following steps are various guidelines for user interface design:

- > The system user should always be aware of what to do next.
- > The screen should be formatted so that various types of information, instructions and messages always appear in the same general display area.
- > Message, instructions or information should be displayed long enough to allow the system user to read them.
- > Use display attributes sparingly.
- > A user should not be allowed to proceed without correcting an error.

Preliminary Product Description:

The first step in the product development cycle is the preliminary investigation to determine the feasibility of the system. The purpose of the preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the business system in all respect. Rather, it is the collection of information that helps committee members to evaluate the merits of the project request and make an informed judgement about the feasibility of the proposed project.

Analysts working on the preliminary investigation should accomplish the following objectives:

- 1) Clarify and understand the project request.
- 2) Determine the size of the project.
- 3) Assess costs and benefits of alternative approaches.
- 4) Determine the technical and operational feasibility of alternative approaches.
- 5) Report the findings to management, with recommendations outlining the acceptance or rejection of the proposal.

The Initial Cost

The initial cost of setting up the system will include the cost of hardware software (OS, add-on software, utilities) & labour (setup & maintenance). The same has to bear by the organization.

The Running Cost

Besides, the initial cost the long term cost will include the running cost for the system including the AMC, stationary charges, cost for human resources, cost for date/renewal of various related software.

Need for Training

The users along with the administrator need to be trained at the time of implementation of the system for smooth running of the system. The client will provide the training site.

Reliable, accurate and secure data was also considered to be complex task without this proposed system. Because there was no such record for keeping track of all the activities, which was done by the Railway Management System on daily basis.

We talked to the management people who were managing at the financial issues of the center, the staff who were keeping the records in lots of registers and the reporting manager regarding their existing system, their requirements abd their expectations from the new proposed system. Then, we did the system study of the entire system based on their requirements and the additional features they wanted to incorporate in this system.

Project Category

Relational Database Management System (RDBMS) is an RDMBS based project which is currently using MySQL for all the transaction statements. MySQL is an opensource RDBMS system.

Brief Introduction about RDBMS

A relational database management system (RDBMS) is a database management system (DBMS) that is based on the relational model as invented by E.F. Codd, of IBM's San Jose Research Laboratory. Many popular databases currently in use are based on the relational database model.

RDBMSs have become a predominant choice for the storage of information in new databases used for financial records, manufacturing and logistical information, personal data, and much more since the 1980s. Relational databases have often replaced legacy hierarchical databases and network databases because they are easier to understand. A DBMS has to be persistent, that is it should be accessible when the program created the data ceases to exist or even the application that created the data restarted. A DBMS also has to provide some uniform methods independent of a specific application for accessing the information that is stored. RDBMS is a Relational Data Base Management System Relational DBMS. This adds the additional condition that the system supports a tabular structure for the data, with enforced relationships between the tables.

Implementation Methodology:

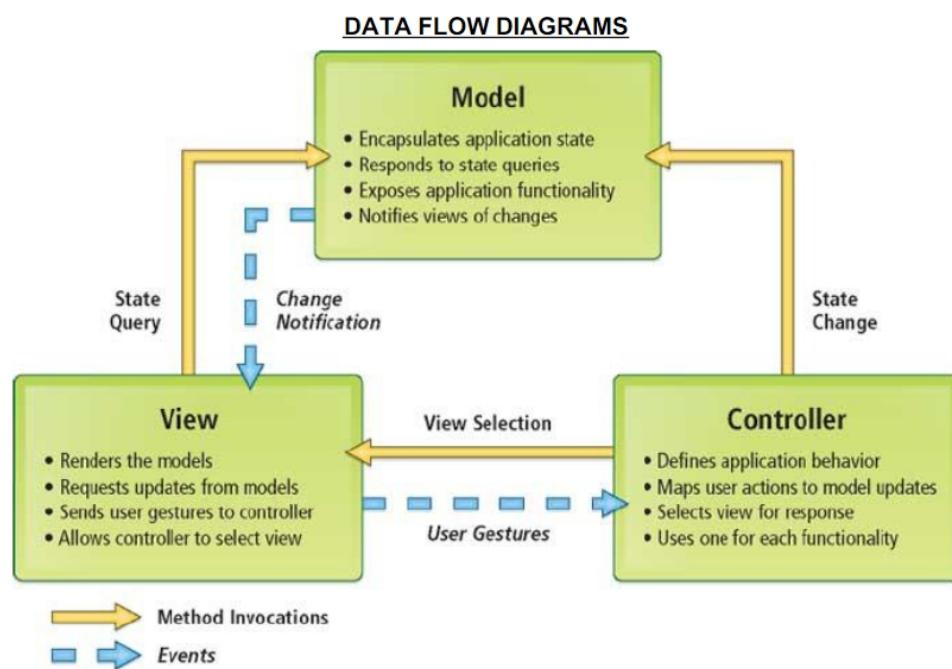
Model View Controller or MVC as it is popularly called, is a software design pattern for developing web applications. A model view controller pattern is made up of the following three parts:

Model: The lowest level of the pattern which is responsible for maintaining data.

View: This is responsible for displaying all or a portion of the data to the user.

Controller: Software Code that controls the interactions between the Model and View.

MVC is popular as it isolates the application logic from the user interface layer and supports separation of concerns. Here the controller receives all requests for the application and then works with the Model to prepare any data needed by the View. The View then uses the data prepared by the Controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows:



3.5 Project Planning

Software project plan can be viewed as the following:

1) Within the Organisation:

How the project is to be implemented? What are the various constraints (Time, Cost, Staff)? What is market strategy?

2) With respect to the Customer:

Weekly or timely meetings with the customer with presentation on status reports. Customers feedback is also taken further modification and developments are done. Project milestones and deliverables are also presented to the customer.

For a successful software project the following steps can be followed:

1) Select a project

- > Identifying the project's aims and objectives
- > Understanding requirements and specifications
- > Methods of analysis, design and implementation.
- > Testing techniques
- > Documentation

2) Project Milestones and deliverables

3) Budget allocation

- > Exceeding limits within control

4) Project Estimates

- > Cost
- > Time
- > Size of Code
- > Duration

5) Resource Allocation

- > Hardware
- > Software
- > Previous relevant project information
- > Digital Library

6) Risk Management

- > Risk avoidance
- > Risk detection

Project Scheduling

An elementary Gantt chart or Timeline for the development plan is given below. The plan explains the tasks versus the time (in weeks) they will take to complete.

| | January | | | | February | | | | March | | | | | | | |
|-----------------------|---------|----|----|----|----------|----|----|----|-------|----|----|----|--|--|--|--|
| Requirement Gathering | | | | | | | | | | | | | | | | |
| Analysis | | | | | | | | | | | | | | | | |
| Design | | | | | | | | | | | | | | | | |
| Coding | | | | | | | | | | | | | | | | |
| Testing | | | | | | | | | | | | | | | | |
| Implement | | | | | | | | | | | | | | | | |
| | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | W1 | W2 | W3 | W4 | | | | |

Cost estimation of the project:

Software cost comprises a small percentage of overall computer-based system cost. There are a number of factors, which are considered, that can affect the ultimate cost of the software such as - human, technical, hardware and software availability etc.

The main point that was considered during the cost estimation of the project was its sizing. In spite of complete software sizing, function point and approximate lines of code were also used to "size" each element of the software and their costing.

The cost estimation done by me for project also depended upon the baseline metrics collected from past projects and these were used in conjunction with estimation variables to develop cost and effort projections.

We have basically estimated this project mainly on two bases -

1) Effort Estimation - This refers to the total man-hours required for the development of the project. It even includes the time required for doing documentation and user manual.

2) Hardware Required Estimation - This includes the cost of the PCs and the hardware cost required for development of this project.

Project Profile:

There has been continuous effort to develop tools, which can ease the process of software development. But, with the evolving trend of different programming paradigms today's software developers are really challenged to deal with the changing technology. Among other issues, software re-engineering is being regarded as an important process in the software development industry.

One of the major tasks here is to understand software systems that are already developed and to transform them to a different software environment. Generally, this requires a lot of manual effort in going through a program that have been developed by another programmer. This project makes a novel attempt to address the issued of program analysis and generation of diagrams, which can depict the structure of a program in a better way.

Today UML is being considered as an individual standard for software engineering design process. It essentially provides several diagramming tools that can express different aspects/characteristics of the program such as:

3.6 Use Cases

Elicit requirement from users in meaningful chunks. Construction planning is built delivering some use cases and each interaction basis for system testing.

Class Diagrams:

Shows static structure of concepts, types and class. Concepts how users think about the world; type shows interfaces of software companies; classes shows implementation of software companies.

Interaction Diagrams:

Shows how several objects collaborate in single use case.

Package Diagram:

Show group of classes and dependencies among them.

State Diagram:

Show how a single object behaves across many use cases.

Activity Diagram:

Shows behavior with control structure. Can show many objects over many uses, many object in single use case, or implementations methods encourage parallel behavior, etc.

The end of this project is a comprehensive tool that can parse any vb.net program and extract most of the object oriented features inherent in the program such as polymorphism, inheritance, encapsulation and abstraction.

What is UML?

UML stands for Unified Modelling Language is the successor to the wave of Object Oriented Analysis & Design (OOA&D) methods that appeared in the late 80's. It is most directly unifies the methods of Booch, Rumbaugh (OMT) and Jacobson. The UML is called a modeling language, not a method. Most methods consist at least in principle, of both a modeling language and a process. The Modeling Language is that notation that methods used to express design.

Notations and meta-methods:

The notation is a graphical stuff; it is the syntax of the modeling language. For instance, class diagram notation defines how many items are concepts such as class, association, and multiplicity is represented. These are:

Class Diagram:

The class diagram technique has become truly central within object oriented methods. Virtually every method has included some variation on this technique.

Association:

It represents between the instances of class. From the conceptual perspective, association represents conceptual relations between classes. Each association has two roles. Each role is a direction on the association. A role also has multiplicity, which is a indication of how many object may participate in the given relationship.

Generalization:

A typical example of generalization evolves the personal and corporate customer of a business. They have differences but also many similarity. The similarities can be placed in generalization with personal customer and corporate customer sub type.

Aggregation:

Aggregation is the part of relationship. It is like saying a car has engine

and wheels as its parts. This sounds good, but difficult thing is considering, what is the difference is aggregation and association.

Interaction:

Interaction diagrams are models that describes how groups of objects collaborate in some behavior.

Typically, an interaction diagram captures the behavior a single use cases.

Package Diagram:

One of the oldest questions in software methods is: how do you break down a large system into smaller systems? It becomes difficult to understand and the changes we make to them.

Structured methods use functional decomposition in which the overall system was mapped as a function broken down into sub function, which is further broken down into sub function and so forth. The separation of process data is gone, functional decomposition is gone, but the old question still remains. One idea is to group the classes together into higher-level unit.

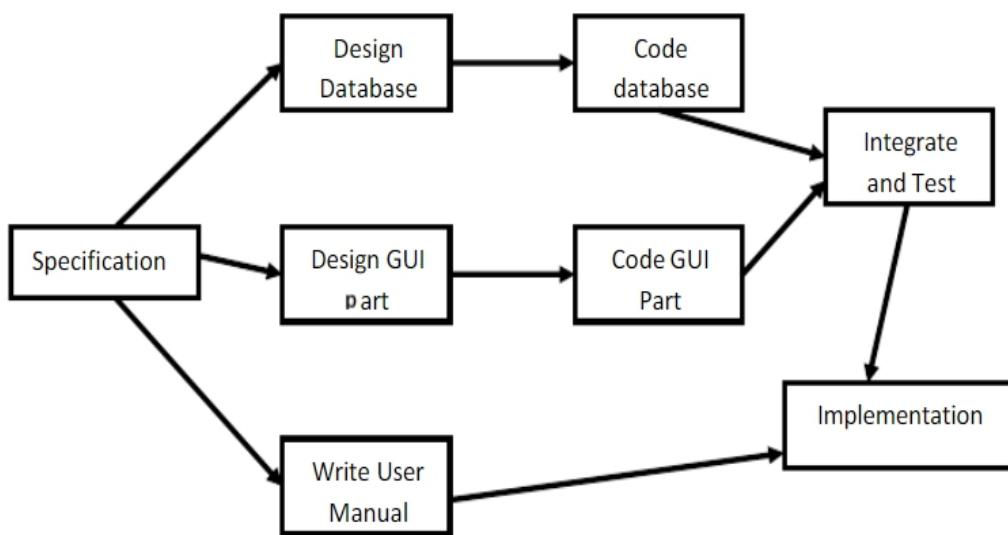
A dependency exists between the two elements if changes to the definition of one element may cause to other. With classes, dependencies exit for various reasons: one class sends a message to another; one class has another as part of its data; one class mentions another as a parameter to an operation. A dependency between two package exists; and any dependencies exist between any two classes in the package.

State Diagram:

State diagram are familiar technique to describe the behavior of a system. They describe all the possible states of a particular object can get into and how the object changes as a result of events that reach the objects. In most OO technique, state diagrams are drawn for a single class to show the lifetime behavior of a single object. There are many forms of state diagram, each with slightly different semantics. The most popular one used in OO technique is based on David Harel's state chart.

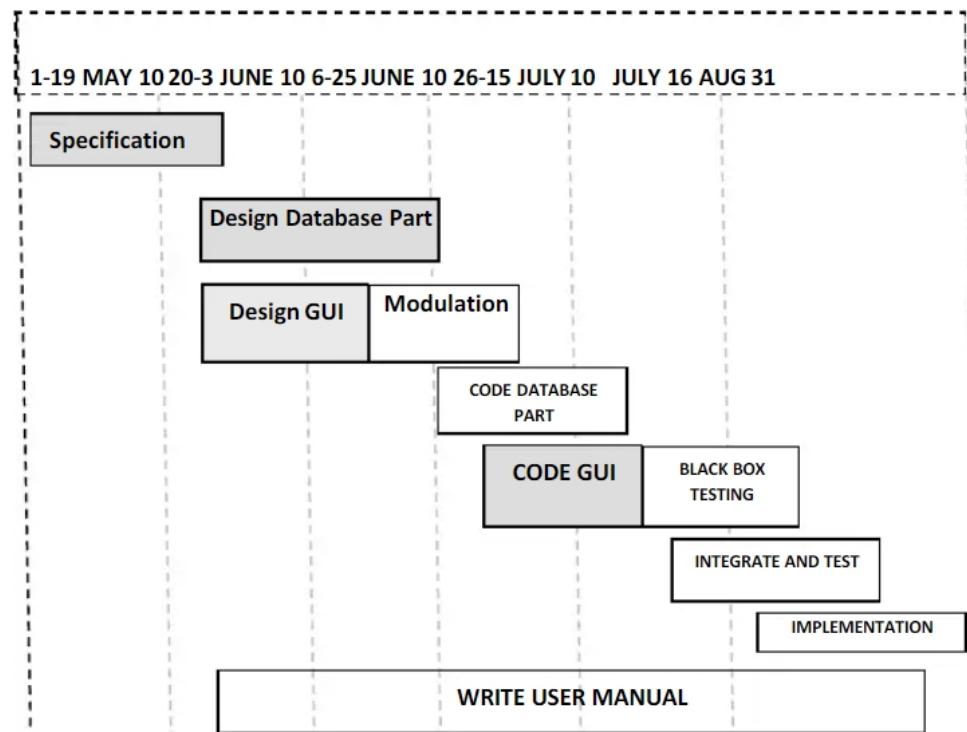
Pert Chart

Is organized for events, activities or tasks. It is a scheduling device that shows graphically the order of the tasks to be performed. It enables the calculation of the critical path. The time and cost associated along a path is calculated and the path requires the greatest amount of elapsed time in critical path.



Gantt Chart

It is also known as Bar Chart and is used exclusively for scheduling purpose. It is a project controlling technique. It is used for scheduling, budgeting and resourcing planning. A Gantt is a bar chart with each bar representing activity. The bars are drawn against a timeline. The length of time planned for the activity. The Gantt chart in the figure shows the gray parts is slack time that is the latest by which a task has been finished.



Use Case Model of the project:

The use case model for any system consists of “Use Cases”. Use cases represent different ways in which the system can be used by the user. A simple way to find all the use case of the system is to ask the questions “What the user can do using this system?”. The user cases partition the system behavior into transactions such that each transaction performs some useful action from the user’s point of view.

The purpose of the use case is to define a piece of coherent behavior without revealing the internal structure of the system. A use case typically represents a sequence of interaction between the user and the system. These interactions consist of one main line sequence represented by the normal interaction between the user and the system. The use case model is an important analysis and design artifact (task). Use cases can be represented by drawing a use case diagram and writing an accompanying text elaborating the drawing.

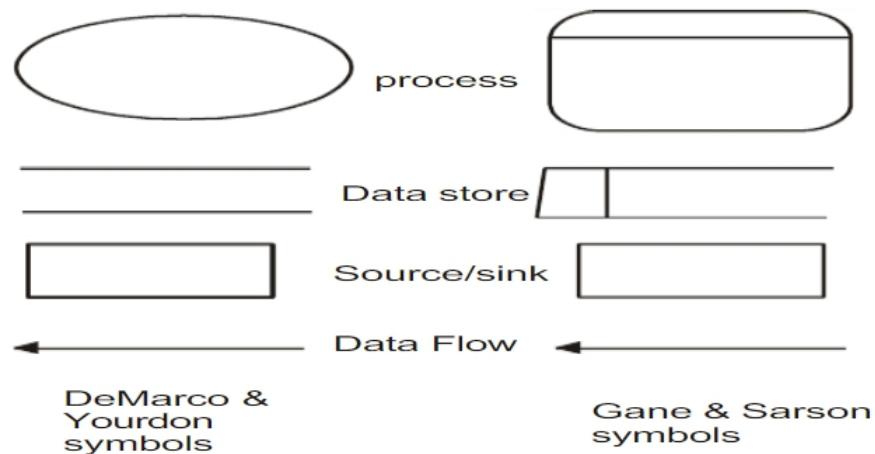
In the use case diagram each use case is represented by an ellipse with the name of each use case written inside the ellipse. All the ellipses of the system are enclosed within a rectangle which represents the system boundary. The name of the system being modeled appears inside the rectangle. The different users of the system are represented by using stick person icon. The stick person icon is normally referred to as an Actor. The line connecting the actor and the use cases is called the communication relationship. When a stick person icon represents an external system it is annotated by the stereotype <<external system>>.

Dataflow Diagram:

It is the starting point of the design phase that functionally decomposes the requirements specification. ADFD consists of a series of bubbles joined by lines. The bubbles represent data transformation and the lines represent data flows in the system. A DFD describes what data flows rather than how they are processed, so it does not have hardware, software and data structure.

A Data-Flow-Diagram (DFD) is a graphical representation of the “flow” of data through an information system. DFDs can also be used for the visualization of data processing (Structured design). A data flow diagram is a significant modeling technique for analyzing and constructing information processes. DFD literally means an illustration that explains the course or movement of information in a process. DFD illustrates this flow of information in a process based on the inputs and outputs. ADFD can be referred to as a Process Model.

Unlike details flow chart, DFDs don't supply detail descriptions of modules that graphically describe a system's data and how the data interact with the system. Data flow diagram number of symbols and the following are of by DeMarco.



There are seven rules for constructing of a dataflow diagram:

- 1) Arrows should not cross each other.
- 2) Squares, circles and files must wear names.
- 3) Decomposed dataflows must be balanced.
- 4) No two dataflows, squares or circles can have the same names.
- 5) Draw all dataflows around outside of the main program.
- 6) Choose meaningful names for dataflows, processes & data stores.
- 7) Control information such as record units, password and validation requirements are not pertinent to a dataflow diagram.

Additionally, a DFD can be utilized to visualize data processing or a structured design.

Chapter - 4

Quick Grocery Store Screenshots

General Modules

Quick Grocery Shopping

localhost:55566/OnlineShop/Home.aspx

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

Cakes & Bakery

| | | | |
|--|--|---|--|
|  |  |  |  |
| Name : Cabbage Price : 17 view | Name : Carrot Price : 62 view | Name : Onion Price : 40 view | Name : Radish Price : 20 view |
|  |  |  |  |
| Name : Ginger Price : 28 view | Name : Mango Price : 68 view | Name : Watermelon Price : 126 view | Name : Apple Price : 159 view |

Category

- Vegetables
- Fruits
- Cakes | Bakes
- Rice | Pulses | Dals
- Frozen Foods
- Biscuits | Chocolates

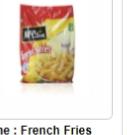
SEARCH

localhost:55566/OnlineShop/PRODUCT.aspx

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

PRODUCTS

| | | | |
|---|---|--|---|
|  |  |  |  |
| Name : French Fries Price : 148 View | Name : Capsicum Price : 33 View | Name : Onion Price : 40 View | Name : Palak Price : 25 View |
|  |  |  | |
| Name : Red Velvet Cake Price : 599 View | Name : Bourbon Chocolate Cream Price : 20 View | Name : Dairy Milk Silk Chocolate Price : 140 View | |

WELCOME



localhost:55566/OnlineShop/Feedback.aspx

QUICK GROCERY

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

LOGIN

Username
Password

LOGIN

[new user](#) [forgot password](#)

Search

CATEGORY

- Vegetables
- Fruits
- Cakes | Bakes
- Rice | Pulses | Dals
- Frozen Foods
- Biscuits | Chocolates

← WELCOME →



FEEDBACK

Name : *
Feedback :

UPLOAD FEEDBACK

localhost:55566/OnlineShop/Registration.aspx

QUICK GROCERY

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

LOGIN

Username
Password

LOGIN

[new user](#) [forgot password](#)

Search

CATEGORY

- Vegetables
- Fruits
- Cakes | Bakes
- Rice | Pulses | Dals
- Frozen Foods
- Biscuits | Chocolates

← WELCOME →



REGISTRATION

First Name : *
Last Name : *
Address : *
City : *
Pincode : *
Gender : Male Female
Mobile :
Email :
Password :
Confirm Password (Same as above)

REGISTER NOW

Already have an account [login here](#)

localhost:55566/OnlineShop/Log

localhost:55566/OnlineShop/Login.aspx

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

LOGIN

Username
Password

LOGIN

[new user](#) [forgot password](#)

Search

CATEGORY

- Vegetables
- Fruits
- Cakes | Bakes
- Rice | Pulses | Dals
- Frozen Foods
- Biscuits | Chocolates

← WELCOME →

USER LOGIN

LOGIN DETAILS

Username :
Password :

LOGIN

Don't have username and Password [register here](#)

localhost:55566/OnlineShop/Con

localhost:55566/OnlineShop/Contact%20us.aspx

HOME PRODUCT FEEDBACK SIGN UP LOGIN CONTACT US

WELCOME TO QUICK GROCERY || ★ || THE ONE STORE FOR ALL YOUR NEEDS

LOGIN

Username
Password

LOGIN

[new user](#) [forgot password](#)

Search

CATEGORY

- Vegetables
- Fruits
- Cakes | Bakes
- Rice | Pulses | Dals
- Frozen Foods
- Biscuits | Chocolates

← WELCOME →

MANAGEMENT

—« • QUICK GROCERY • »—

| COMPANY | FOUNDER | FOUNDER |
|---------|---------|---------|
| | | |

ESTABLISHMENT 2021 Name Aditya VGP Name Bronson T

CONTACT TEL 25 / 29062021 Address Sector 2, Vinayakanagar, Bangalore, 560017 Address Sector 5, Marathahalli, Bangalore, 560103
FAX (222 - 5679455)
+91 1236454631
+91 6450113585

CITY Bangalore Contact No +91 8971880436 Contact No +91 8095807528

User Modules (Ex):

The screenshot shows a web browser window for 'localhost:55566/OnlineShop/User/Default.aspx'. The title bar says 'localhost:55566/OnlineShop/User/Default.aspx'. The page header includes a logo for 'QUICK GROCERY', a search bar, and a welcome message 'Welcome : Aditya Venkat Ganesh P' with a 'LOGOUT' link. Below the header is a navigation menu with links for Home, My Account, My Order, My Payment, My Cart, and Change Password. The main content area is titled 'PRODUCTS' and displays a grid of 12 vegetable items, each with an image, name, price, and a 'view' link.

| PRODUCTS | | | |
|---|--|---|--|
| Name : Cabbage Price : 17 view | Name : Carrot Price : 62 view | Name : Onion Price : 40 view | Name : Radish Price : 20 view |
| Name : Ginger Price : 28 view | Name : Mango Price : 68 view | Name : Watermelon Price : 126 view | Name : Apple Price : 159 view |
| Name : Ladies Finger | Name : Potato | Name : Capsicum | Name : Palak |

The screenshot shows a web browser window for 'localhost:55566/OnlineShop/User/MyAccount.aspx'. The title bar says 'localhost:55566/OnlineShop/User/MyAccount.aspx'. The page header includes a logo for 'QUICK GROCERY', a search bar, and a welcome message 'Welcome : Aditya Venkat Ganesh P' with a 'LOGOUT' link. Below the header is a navigation menu with links for Home, My Account, My Order, My Payment, My Cart, and Change Password. The main content area is titled 'UPDATE PROFILE' and contains a form with fields for Name, Last name, Address, City, Pincode, and Mobile, each with a corresponding input field. A 'UPDATE PROFILE' button is at the bottom of the form.

S localhost:55566/OnlineShop/User/MyOrder.aspx +

localhost:55566/OnlineShop/User/MyOrder.aspx

Apps Gmail YouTube YouTube on TV Kadokawa - Anime... All Games by [Geni... Other bookmarks Reading list

QUICK GROCERY **SEARCH**

Welcome : Aditya Venkat Ganesh P [LOGOUT](#)

Home My Account My Order My Payment My Cart Change Password

MY ORDERS

| IMAGE | PRODUCT | QUANTITY | ITEM PRICE | TOTAL PRICE |
|-------|---------------|----------|------------|-------------|
| | Vanilla Cake | 1 | 499 | 499 |
| | Cabbage | 1 | 17 | 17 |
| | Onion | 1 | 40 | 40 |
| | Crispy Smiles | 1 | 257 | 257 |

S localhost:55566/OnlineShop/User/MyPayment.aspx +

localhost:55566/OnlineShop/User/MyPayment.aspx

Apps Gmail YouTube YouTube on TV Kadokawa - Anime... All Games by [Geni... Other bookmarks Reading list

QUICK GROCERY **SEARCH**

Welcome : Aditya Venkat Ganesh P [LOGOUT](#)

Home My Account My Order My Payment My Cart Change Password

MY PAYMENT

| PAYMENT TYPE | BANK NAME | BANK BRANCH | CARD NO | CVV NO | AMOUNT | DATE |
|------------------|-----------|-------------|------------------|--------|--------|---------------------|
| CREDIT CARD | KOTAK | CHENNAI | 5648269851683213 | 541 | 499 | 10-08-2021 20:19:20 |
| CASH ON DELIVERY | | | | 0 | 17 | 10-08-2021 20:26:43 |
| CASH ON DELIVERY | | | | 0 | 40 | 10-08-2021 20:31:09 |

The screenshot shows a web browser window for 'localhost:55566/OnlineShop/User/MyCart.aspx'. The title bar says 'localhost:55566/OnlineShop/User/MyCart.aspx'. The page header includes a logo for 'QUICK GROCERY', a search bar, and a welcome message 'Welcome : Aditya Venkat Ganesh P' with a 'LOGOUT' link. Below the header is a navigation menu with links: Home, My Account, My Order, My Payment, My Cart, and Change Password. The main content area is titled 'QUICK GROCERY CART' and contains a table with three rows of items. The table columns are: IMAGE, PRODUCT, PRICE, QUANTITY, TOTAL PRICE, UPDATE, and CANCEL. The items are:

| IMAGE | PRODUCT | PRICE | QUANTITY | TOTAL PRICE | UPDATE | CANCEL |
|-------|-------------------------|-------|--------------------------------|-------------|------------------------|------------------------|
| | Chocolate Donut | 80 | <input type="text" value="1"/> | 80 | update | cancel |
| | Dark Fantasy | 80 | <input type="text" value="2"/> | 160 | update | cancel |
| | Bourbon Chocolate Cream | 20 | <input type="text" value="1"/> | 20 | update | cancel |

At the bottom of the cart area are two buttons: 'CONTINUE BROWSING' and 'CHECK OUT'.

Admin Modules (Ex):

The screenshot shows a web browser window for 'localhost:55566/OnlineShop/admin/Addcategory.aspx'. The title bar says 'localhost:55566/OnlineShop/admin/Addcategory.aspx'. The page header includes a logo for 'QUICK GROCERY' and the text 'QUICK GROCERY STORE'. On the left is a vertical sidebar with links: Add Category, Add Product, Add Quantity, View Order, View Payment, Reports, FeedBack, Change Password, and Log Out. The main content area has a title 'ADD CATEGORY' and a form for adding a category. It includes a text input field 'Category name : ', a red 'ADD' button, and a table showing existing categories. The table columns are: Delete, Edit, and CATEGORY NAME. The data is:

| Delete | Edit | CATEGORY NAME |
|------------------------|----------------------|-----------------------|
| Delete | Edit | Vegetables |
| Delete | Edit | Fruits |
| Delete | Edit | Cakes Bakes |
| Delete | Edit | Rice Pulses Dals |
| Delete | Edit | Frozen Foods |
| Delete | Edit | Biscuits Chocolates |

localhost:55566/OnlineShop/admin

QUICK GROCERY STORE

ADD PRODUCT

| | |
|-----------------|--------------------------------------|
| Add Category | Item name : <input type="text"/> |
| Add Product | Description : <input type="text"/> |
| Add Quantity | Price : <input type="text"/> |
| View Order | Quantity : <input type="text"/> |
| View Payment | Image : <input type="file"/> |
| Reports | Image 1 : <input type="file"/> |
| FeedBack | Image 2 : <input type="file"/> |
| Change Password | Discount : <input type="text"/> |
| Log Out | Category name : <input type="text"/> |

ADD ITEM

| | ITEM NAME | PRICE |
|--------|------------|-------|
| Delete | Cabbage | 17 |
| Delete | Carrot | 62 |
| Delete | Onion | 40 |
| Delete | Radish | 20 |
| Delete | Ginger | 28 |
| Delete | Mango | 68 |
| Delete | Watermelon | 126 |

localhost:55566/OnlineShop/admin

QUICK GROCERY STORE

ADD QUANTITY

| | | | | |
|-------------------------------------|---------------|----------------|--------------------|---------------|
| Select item : <input type="text"/> | ITEM NAME | TOTAL QUANTITY | AVAILABLE QUANTITY | SELL QUANTITY |
| Add quantity : <input type="text"/> | Cabbage | 1 | 0 | 3 |
| | Carrot | 1 | 0 | 1 |
| | Onion | 1 | 0 | 1 |
| | Radish | 1 | | |
| | Ginger | 1 | | |
| | Mango | 2 | 1 | 1 |
| | Watermelon | 1 | 1 | 1 |
| | Apple | 3 | | |
| | Ladies Finger | 1 | | |
| | Potato | 3 | | |
| | Capsicum | 2 | | |
| | Palak | 1 | | |
| | Corn | 2 | 1 | 1 |
| | Pomegranate | 2 | | |

localhost:55566/OnlineShop/admin +

localhost:55566/OnlineShop/admin/VIEWORDER.aspx

Apps Gmail YouTube YouTube on TV Kadokawa - Anime... All Games by [Geni... Other bookmarks Reading list

QUICK GROCERY STORE

| VIEW ORDERS | | | | |
|---------------------------------|-------|--------------|----------|-------|
| | IMAGE | PRODUCT | QUANTITY | PRICE |
| Add Category | | Vanilla Cake | 1 | 499 |
| Add Product | | Cabbage | 1 | 17 |
| Add Quantity | | Onion | 1 | 40 |
| View Order | | | | |
| View Payment | | | | |
| Reports | | | | |
| FeedBack | | | | |
| Change Password | | | | |
| Log Out | | | | |

localhost:55566/OnlineShop/admin +

localhost:55566/OnlineShop/admin/VIEWPAYMENT.aspx

Apps Gmail YouTube YouTube on TV Kadokawa - Anime... All Games by [Geni... Other bookmarks Reading list

QUICK GROCERY STORE

| PAYMENT REPORTS | | | | | | |
|------------------|-----------|-------------|------------------|--------|--------|---------------------|
| PAYMENT TYPE | BANK NAME | BANK BRANCH | CARD No | CVV No | AMOUNT | PAYMENT DATE |
| CREDIT CARD | KOTAK | CHENNAI | 5648269851683213 | 541 | 499 | 10-08-2021 20:19:20 |
| CASH ON DELIVERY | | | | 0 | 17 | 10-08-2021 20:26:43 |
| CASH ON DELIVERY | | | | 0 | 40 | 10-08-2021 20:31:09 |

| REPORTS | | | | | |
|--|-------------|-------|----------|-----------|---------|
| Select category : Fruits <input type="button" value="VIEW"/> | | | | | |
| Total 10 Record Found | | | | | |
| IMAGE | PRODUCT | PRICE | QUANTITY | AVAILABLE | SELLQNT |
| | Mango | 68 | 2 | 1 | 1 |
| | Watermelon | 126 | 1 | 1 | 1 |
| | Apple | 159 | 3 | 3 | 3 |
| | Pomegranate | 139 | 3 | 3 | 3 |
| | Banana | 66 | 1 | 1 | 1 |
| | Orange | 125 | 1 | 1 | 1 |
| | Avocado | 119 | 2 | 2 | 2 |

Back-end Functioning:



Microsoft SQL Server Management Studio

File Edit View Query Debug Tools Window Community Help

New Query Execute

Object Explorer

Connect master

Databases

- System Databases
- OnlineShop
- Database Diagrams
- Tables
- System Tables
- dbo.AdminMst
- dbo.CateMst
- dbo.Feedbackmstr
- dbo.ItemMst
- dbo.OrderMst
- dbo.PaymentMst
- dbo.UserMst
- Views
- Synonyms
- Programmability
- Stored Procedures
- dbo.ADMIN_CHANGE_F
- dbo.ADMIN_DELETE
- dbo.ADMIN_INSERT
- dbo.ADMIN_LOGIN
- dbo.ADMIN_SELECT
- dbo.CATEMST_DELETE
- dbo.CATEMST_INSERT
- dbo.CATEMST_SELECT
- dbo.CATEMST_UPADAT
- dbo.FEEDBACKMSTR_DI
- dbo.FEEDBACKMSTR_IN
- dbo.FEEDBACKMSTR_SE
- dbo.FEEDBACKMSTR_UI
- dbo.ITEM_DELETE
- dbo.ITEM_INSERT
- dbo.ITEM_SELECT
- dbo.ITEM_SELECT_BY_C
- dbo.ITEM_SELECT_FOR_

SQLQuery1.sql - LAPTOP-PNOM7HH...

```

USE [OnlineShop]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ItemMst] (
    [IID] [int] IDENTITY(1,1) NOT NULL,
    [IName] [nvarchar](50) NULL,
    [Detail] [nvarchar](500) NULL,
    [Price] [float] NULL,
    [Image] [nvarchar](50) NULL,
    [Qnt] [int] NULL,
    [AQnt] [int] NULL,
    [SQnt] [int] NULL,
    [CName] [nvarchar](50) NULL,
    [EntryDate] [datetime] NULL,
    [Image1] [nvarchar](500) NULL,
    [Image2] [nvarchar](500) NULL,
    [Size] [int] NULL,
    CONSTRAINT [PK_ItemMst] PRIMARY KEY CLUSTERED
(
    [IID] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CateMst] (
    [CID] [int] IDENTITY(1,1) NOT NULL,
    [CName] [nvarchar](50) NULL,
    CONSTRAINT [PK_CateMst] PRIMARY KEY CLUSTERED
(
    [CID] ASC
)

```

Connected. (1/1) | LAPTOP-PNOM7HH\SQLEXPRESS... | LAPTOP-PNOM7HH\アディティア... | master | 00:00:00 | 0 rows | Ln 13 Col 26 Ch 23 | INS

Ready

4.1 Example Source Code of HTML:

```
<%@ Page Title="" Language="C#" MasterPageFile "~/User.master" AutoEventWireup="true"
CodeFile="Login.aspx.cs" Inherits="Login" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" Runat="Server">
    <style type="text/css">
        .style2
        {
            width: 259px;
        }
        .style3
        {
            height: 8px;
        }
        .style4
        {
            height: 9px;
        }
        .style5
        {
            width: 100px;
        }
        .style6
        {
            width: 154px;
        }
        .style7
        {
            height: 9px;
            width: 154px;
        }
        .style8
        {
            width: 154px;
            height: 8px;
        }
        .style9
        {
            height: 343px;
            width: 820px;
        }
        .style10
        {
            height: 30px;
            color: #fff;
            text-align: center;
            font-weight: bold;
            background: green;
            padding: 2px;
            font-size: 14px;
            width: 820px;
        }
    </style>
</asp:Content>
```

```

.style11
{
    width: 820px;
}
</style>
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1" Runat="Server">
    <table class="tbl">
        <tr>
            <td class="style10">
                USER LOGIN </td>
        </tr>
        <tr>
            <td class="style11">
                &nbsp;</td>
        </tr>
        <tr>
            <td class="style11">
                <br />
                <br />
            </td>
        </tr>
        <tr>
            <td style="text-align: center" class="style9">
                <table align="center" class="style2" style="border: thin solid #008000">
                    <tr>
                        <td class="tblhead" colspan="2">
                            LOGIN DETAILS</td>
                    </tr>
                    <tr>
                        <td>
                            &nbsp;</td>
                        <td class="style6">
                            &nbsp;</td>
                    </tr>
                    <tr>
                        <td class="lbl">
                            Username :
                        </td>
                        <td class="style6">
                            <asp:TextBox ID="txtuname" runat="server" CssClass="txt"
Width="140px"></asp:TextBox>
                            <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"
                                ControlToValidate="txtuname" ErrorMessage="Enter
Username"
                                ForeColor="#666666"></asp:RequiredFieldValidator>
                        </td>
                    </tr>
                    <tr>
                        <td class="lbl">
                            Password :</td>
                        <td class="style6">
                            <asp:TextBox ID="txtpass" runat="server" CssClass="txt"
Width="140px"
                                TextMode="Password"></asp:TextBox>
                        </td>
                    </tr>
                </table>
            </td>
        </tr>
    </table>
</asp:Content>

```

```

<tr>
    <td class="style4">
    </td>
    <td class="style7">
        <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server"
            ControlToValidate="txtpass" ErrorMessage="Enter Password"
            ForeColor="#666666"></asp:RequiredFieldValidator>
    </td>
</tr>
<tr>
    <td>
        &nbsp;</td>
    <td align="left" valign="top" class="style6">
        <table class="style5">
            <tr>
                <td>
                    <asp:Button ID="btnlogin" runat="server"
CssClass="btn" Text="Login"
Width="105px" />
                </td>
                <td>
                    &nbsp;</td>
                </tr>
            </table>
        </td>
    </tr>
    <tr>
        <td class="style3">
        </td>
        <td align="left" class="style8">
            &nbsp;
            <asp:Label ID="lblerror" runat="server"
ForeColor="#666666"></asp:Label>
            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
            </td>
        </tr>
    </table>
</td>
</tr>
<tr>
    <td style="text-align: center; color: Green" class="style11">
        Don't have username and Password
        <asp:HyperLink ID="HyperLink2" runat="server"
NavigateUrl "~/Registration.aspx"
            ForeColor="#0099FF">register here</asp:HyperLink>
    </td>
</tr>
</table>
</asp:Content>

```

Example Source Code of C#:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class User_MyCart : System.Web.UI.Page
{
    DS_ORDER.ORDER_SELECTDataTable ODT = new DS_ORDER.ORDER_SELECTDataTable();
    DS_ORDERTableAdapters.ORDER_SELECTTableAdapter OAdapter = new
    DS_ORDERTableAdapters.ORDER_SELECTTableAdapter();
    DS_AITEM.ITEM_SELECTDataTable IDT = new DS_AITEM.ITEM_SELECTDataTable();
    DS_AITEMTableAdapters.ITEM_SELECTTableAdapter IAdapter = new
    DS_AITEMTableAdapters.ITEM_SELECTTableAdapter();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (Page.IsPostBack == false)
        {

            ODT = OAdapter.Sleect_BY_UNAME_Status(Session["uname"].ToString(), 0);
            GridView1.DataSource = ODT;
            GridView1.DataBind();
        }
    }
    protected void GridView1_SelectedIndexChanged(object sender, EventArgs e)
    {

    }
    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        int oidd=Convert.ToInt32( GridView1.DataKeys[e.RowIndex].Value);

        OAdapter.Delete(oidd);

        ODT = OAdapter.Sleect_BY_UNAME_Status(Session["uname"].ToString(), 0);
        GridView1.DataSource = ODT;
        GridView1.DataBind();
    }
    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        int oidd=Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
        TextBox txtqq=GridView1.Rows[e.RowIndex].Cells[3].FindControl("txtq") as
        TextBox;
        string pricee = GridView1.Rows[e.RowIndex].Cells[2].Text;
        double tpricee=Convert.ToInt32(pricee) * Convert.ToInt32(txtqq.Text);
        OAdapter.ORDER_UPDATE_CART_ALREADY_ADD(oidd, Convert.ToInt32(txtqq.Text),
        tpricee);

        ODT = OAdapter.Sleect_BY_UNAME_Status(Session["uname"].ToString(), 0);
        GridView1.DataSource = ODT;
    }
}

```

```
GridView1.DataBind();
}
protected void GridView1_RowCommand(object sender, GridViewCommandEventArgs e)
{
}
protected void Button3_Click(object sender, EventArgs e)
{
    ODT = OAdapter.Sleect_BY_UNAME_Status(Session["uname"].ToString(), 0);
    for(int i=0; i<ODT.Rows.Count; i++)
    {
        IDT = IAdapter.Select_BY_INAME(ODT.Rows[i]["iname"].ToString());
        if (Convert.ToInt32(IDT.Rows[0]["Aqnt"].ToString()) <
Convert.ToInt32(ODT.Rows[i]["Qnt"].ToString()))
        {
            lblmsg.Text = ODT.Rows[i]["iname"].ToString() + " Quantity must be
less than purchase quantity";
        }
        else
        {
            Response.Redirect("Checkout.aspx");
        }
    }
}

}
protected void Button2_Click(object sender, EventArgs e)
{
}
}
```

Example Source Code of MySQL:

```

USE [OnlineShop]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ItemMst](
    [IID] [int] IDENTITY(1,1) NOT NULL,
    [IName] [nvarchar](50) NULL,
    [Detail] [nvarchar](500) NULL,
    [Price] [float] NULL,
    [Image] [nvarchar](50) NULL,
    [Qnt] [int] NULL,
    [AQnt] [int] NULL,
    [SQnt] [int] NULL,
    [CName] [nvarchar](50) NULL,
    [EntryDate] [datetime] NULL,
    [Image1] [nvarchar](500) NULL,
    [Image2] [nvarchar](500) NULL,
    [Size] [int] NULL,
    CONSTRAINT [PK_ItemMst] PRIMARY KEY CLUSTERED
(
    [IID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CateMst](
    [CID] [int] IDENTITY(1,1) NOT NULL,
    [CName] [nvarchar](50) NULL,
    CONSTRAINT [PK_CateMst] PRIMARY KEY CLUSTERED
(
    [CID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TABLE [dbo].[AdminMst] (
    [IID] [int] IDENTITY(1,1) NOT NULL,
    [Username] [nvarchar](50) NULL,
    [Password] [nvarchar](50) NULL,
    CONSTRAINT [PK_AdminMst] PRIMARY KEY CLUSTERED
(
    [IID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Feedbackmstr] (
    [FID] [int] IDENTITY(1,1) NOT NULL,
    [UNAME] [nvarchar](50) NULL,
    [MESSAGE] [nvarchar](50) NULL,
    [ENTRYDATE] [datetime] NULL,
    CONSTRAINT [PK_Feedbackmstr] PRIMARY KEY CLUSTERED
(
    [FID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[OrderMst] (
    [OID] [int] IDENTITY(1,1) NOT NULL,
    [Uname] [nvarchar](50) NULL,
    [Iname] [nvarchar](50) NULL,
    [Qnt] [int] NULL,
    [Price] [float] NULL,
    [TPrice] [float] NULL,
    [Status] [int] NULL,
    [EntryDate] [datetime] NULL,
    [image] [nvarchar](500) NULL,
    CONSTRAINT [PK_OrderMst] PRIMARY KEY CLUSTERED
(
    [OID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

create procedure ORDER_SELECT_distinct_uname
as
begin
set nocount on;
select distinct OID,Uname from OrderMst with(nolock)
end
go

-----
--for user registration
-----
create procedure USER_INSERT
(
@Name nvarchar(50),
@Surname nvarchar(50),
@ADD nvarchar(50),
@City nvarchar(50),
@PIN nvarchar(50),
@MOB nvarchar(50),
@email nvarchar(50),
@PASS nvarchar(50)

)
as
begin
insert into
UserMst(Name, Surname, Address, City, Pincode, Mobile, Email, Password, EntryDate)
values (@Name, @Surname, @ADD, @City, @PIN, @MOB, @Email, @PASS, GETDATE())
end
go

alter procedure USER_INSERT
(
@Name nvarchar(50),
@Surname nvarchar(50),
@ADD nvarchar(50),
@City nvarchar(50),
@PIN nvarchar(50),
@MOB nvarchar(50),
@email nvarchar(50),
@PASS nvarchar(50)

)
as
begin
insert into
UserMst(Name, Surname, Address, City, Pincode, Mobile, Email, Password, EntryDate)
values (@Name, @Surname, @ADD, @City, @PIN, @MOB, @Email, @PASS, GETDATE())
end
go
-----
create procedure dbo.USER_SELECT_by_email
(
@email nvarchar(50)
)
as
begin

```

```

select * from UserMst where Email=@email;
end
go

-----
create procedure dbo.USER_UPDATE
(
@UID int,
@Name nvarchar(50),
@Surname nvarchar(50),
@ADD nvarchar(50),
@City nvarchar(50),
@PIN nvarchar(50),
@MOB nvarchar(50)
)
as
begin
update UserMst set
Name=@Name, Surname=@Surname, Address=@ADD,City=@City, Pincode=@PIN, Mobile=@MOB,EntryDate=GETDATE() Where UID=@UID;
end
go

alter procedure dbo.USER_UPDATE
(
@UID int,
@Name nvarchar(50),
@Surname nvarchar(50),
@ADD nvarchar(50),
@City nvarchar(50),
@PIN nvarchar(50),
@MOB nvarchar(50)
)
as
begin
update UserMst set
Name=@Name, Surname=@Surname, Address=@ADD,City=@City, Pincode=@PIN, Mobile=@MOB,EntryDate=GETDATE() Where UID=@UID;
end
go
-----

create procedure dbo.ORDER_SELECT_BY_UNAME
(
@UNAME nvarchar(50)
)
as
begin
select * from OrderMst where Uname=UNAME
End
go

-----
create procedure PAYMENTMST_SELECT_BY_Uname
(
@UNAME nvarchar(50)
)

```

```

as
begin
set nocount on;
select * from PaymentMst with(nolock) where Uname=@UNAME;
end

-----
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure ORDER_SELECT_BY_Uname_ITEM
(
@UNAME nvarchar,
@INAME nvarchar(50)
)
as
begin
set nocount on;
SELECT * FROM OrderMst where Uname=@UNAME AND Iname=@INAME;

end
go

alter procedure ORDER_SELECT_BY_Uname_ITEM
(
@UNAME nvarchar,
@INAME nvarchar(50)
)
as
begin
set nocount on;
SELECT * FROM OrderMst where Uname=@UNAME AND Iname=@INAME;

end
go

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure ORDER_SELECT_BY_Uname_Status
(
@UNAME nvarchar(50),
@Status int
)
as
begin
SELECT * FROM OrderMst where Uname=@UNAME AND Status=@Status
End

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

alter procedure ORDER_SELECT_BY_Status
(
@UNAME nvarchar(50),
@Status int
)
as
begin
SELECT * FROM OrderMst where Uname=@UNAME AND Status=@Status
End

-----
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
create procedure ORDER_SELECT_TOTAL_AMT_PAY
(
@UNAME nvarchar(50)
)
as
begin
SELECT * FROM OrderMst where Uname=@UNAME
end
go

-----
create procedure PAYMENTMST_INSERT
(
@Uname nvarchar(50),
@Amount decimal(18,2),
@Type nvarchar(50),
@Bank nvarchar(50),
@Branch nvarchar(50),
@CardNo nvarchar(50),
@CCV int,
@EntryDate datetime
)
as

begin
insert into PaymentMst
(Uname,Amount,Type,Bank,Branch,CardNo,CCV,EntryDate)
values (@Uname,@Amount,@Type,@Bank,@Branch,@CardNo,@CCV,GETDATE())
end

go
-----
select * from ItemMst

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create procedure ITEM_UPDATE_SELL
(
@INAME nvarchar(50),
@QNT int
)

```

```

as
begin
update ItemMst set AQnt=Qnt-@QNT, SQnt=isnull(SQnt, 0)+@QNT where
IName=@INAME
end
go

-----
create procedure ITEM_UP
(
@IID int,
@INAME nvarchar(50),
@PRICE decimal(10,2)
)
as
begin
Update ItemMst set IName=@INAME, Price=@PRICE where IID=@IID;
end
-----

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create procedure ORDER_UPDATE_STATUS_BY_NAME
(
@UNAME nvarchar(50),
@STATUS int
)
as
begin
Update OrderMst set Status=@STATUS where Uname=@UNAME;
end
go
-----

create procedure PAYMENT_SELECT_for_distinct_uname

as
begin
SELECT distinct Uname FROM PaymentMst
end
go
-----

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

alter procedure PAYMENT_SELECT_for_distinct_uname

as
begin
SELECT distinct Uname FROM PaymentMst
end
go
-----
```

```

create procedure USER_SELECT_FORGOT_PASS
(
@email nvarchar(50),
@Mobile nvarchar(50)
)
as
begin
set nocount on;
Select * from UserMst where Email=@email AND Mobile=@Mobile
end
go

create procedure dbo.ORDER_DELETE
(
@ID int
)
as
begin
Delete from OrderMst where OID=@ID
end

-----
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

create procedure dbo.ORDER_UPDATE_CART_ALREADY_ADD
(
@OID int,
@Qnt int,
@TPrice int
)
as
begin

Update OrderMst set Qnt=@Qnt,TPrice=Price*@Qnt Where OID=@OID
end

-----
insert into AdminMst(Username,Password)values('Aditya','Chittu2000')
insert into UserMst(Name,Password)values('Aditya','Chittu2000')

SELECT * FROM UserMst
SELECT * FROM CateMst
SELECT * FROM ItemMst
SELECT * FROM OrderMst
SELECT * FROM PaymentMst
SELECT * FROM AdminMst
select * from Feedbackmstr

```

Testing

5.1 Introduction to Testing

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use. Software testing involves the execution of a software component or system component to evaluate one or more properties of interest.

Security Testing of the Project:

Testing is vital for the success of any software. No system design is ever perfect. Testing is also carried in two phases. First phase is during the software engineering that is during the module creation. Second phase is after the completion of software, this is system testing which verifies that the whole set of programs hanged together.

5.2 Types of Testing:

White Box Testing:

In this technique, the close examination of the logical parts through the software are tested by cases that exercise specific sets of conditions or loops, all logical parts of the software is checked once. Errors that can be corrected using this technique are typographical errors, logical expressions which should be executed once may be getting executed more than once and error resulting by using wrong controls and loops.

White-box testing, sometimes called Glass box testing or structural testing or clear box testing. White box testing is used to test areas that cannot be reached from a black box testing. White-box testing user an internal perspective of the system to design the test cases based on the internal structure.

Black Box Testing:

This method enables the software engineer to devise sets of input techniques that fully exercise all functional requirements for a program. Black box testing tests the input, the output and the external data. It checks whether the input data is incorrect and whether we are getting the desired output.

It focuses on the functional requirements of the software. In black-box testing, the cases are derived from the specification of the system, the implementation details of the system are not taken into considerations.

Alpha Testing:

Acceptance testing is also sometimes called alpha testing. Bespoke systems are developed for a single customer. The alpha testing proceeds until the system developer and the customer agree that the provided system is acceptable implementation of the system requirements.

Beta Testing:

On the other hand, when a system is to be marked as a software product, another process called beta testing is often conducted. During beta testing, a system is delivered among a number of potential users who agree to use it. The customers then report problems to the developers. This provides the product for real use and detects errors which may not have been anticipated by the system developers.

Cleanroom Testing:

Clean room testing makes use of incremental development model. The initial increment delivered to the customer with the critical functionalities and less important features are added in later increments. The initial increment is released for experimentation with an intention to identify the requirements problem. These problems are removed in later increments. New increments are added to the existing ones and the integrated system is validated. As new increments are added, the initial increments are also released.

Unit Testing:

Each module is considered independently. It focuses on each unit of software as implemented in the source code. It is White box testing.

Integration Testing:

It aims at constructing the program structure while at the same constructing tests to uncover errors associated with interacting the modules. Modules are integrated by using top-down approach.

Validation Testing:

Preformed to ensure all functional and performance requirements are met.

System Testing:

It is executing the programs to check logical changes made in it with intention of finding errors. A system is tested for online response, volume of transaction, recovery from failure etc. System testing is done to ensure that the system satisfies all the user requirements.

Implementation and Software Specification Testings:

Detailed Design of Implementation:

This phase of the systems development life cycle refines hardware and software specifications, establishes programming plans, train users and implements extensive testing procedures, to evaluate design and operating specifications and/or provide the basis for further modification.

Technical Design:

This activity builds upon specifications produced during new system design, adding detailed technical specifications and documentation.

Test Specifications and Planning:

This activity prepares detailed test specifications for individual modules and programs, job streams, subsystems, and for the system as a whole.

Programming and Testing:

This activity encompasses actual development, writing and testing of program units or modules.

User Training:

This activity encompasses writing user procedure manuals, preparation of user training materials, conducting training programs, and testing procedures.

Acceptance Testing:

A final procedural review to demonstrate a system and secure user approval before a system becomes operational.

Installation Phase:

In this phase the new computerized system is installed, the conversion to new procedures is fully implemented, and the potential of the new system is explored.

System Installation:

The process of starting the actual use of a system and training user personal in its operation.

Review Phase:

This phase evaluates the success and failures during a systems development project, and to measure the results of a new computerized transystem in terms of benefits and savings projected at the start of the project.

Development Recap:

A review of a project immediately after completion to find successes and potential problems in future work.

Post-Implementation review:

A review, conducted after a new system has in operation for sometime, to evaluate actual system performance against original expectations and projections for cost-benefit improvements. Also identifies maintenance projects to enhance or improve the system.

The steps involved in the Software Testing:

- a. Preparation of the test cases.
- b. Preparation of the possible test data with all the validation checks.
- c. Complete code review of the module.
- d. Actual testing done manually.
- e. Modifications done for the errors found during testing.
- f. Prepared the test result scripts.

The Unit testing done included the testing of the following items:

1. Functionality of the entire module/forms.
2. Validations for user input.
3. Checking of the Coding standards to be maintained during coding.
4. Testing the module with all the possible test data.
5. Testing of the functionality involving all type of calculations etc.
6. Commenting standard in the source files.

After completing the unit testing of all the modules, the whole system is integrated with all its dependencies in that module. While system Integration, we integrated the modules one by one and tested the system at each step. This helped in reduction of errors at the time of the system testing.

The steps involved during systems testing are as follows:

- > Integration of all the modules/forms in the system.
- > Preparation of the test cases.
- > Preparation of the possible test data with all the validation checks.
- > Actual testing done manually.
- > Recording of all the reproduced errors.
- > Modifications done for the errors found during testing.
- > Prepared the test result scrpts after rectification of the errors.

The System Testing done included the testing of the following items:

- 1) Functionality of the entire system as a whole.
- 2) User interface of the system.
- 3) Testing the dependent modules together with all the possible test data scripts.
- 4) Verification and Validation testing.
- 5) Testing the reports with all its functionality.

After the completion of the system testing, the next following phase was the acceptance testing. Clients at their end did this and accepted the system with appreciation. Thus, we reached the final phase of the project delivery.

Conclusion

6.1 Conclusion of the project

The project entitled **Quick Grocery Store** was completed successfully. The system has been developed with much care and free of errors and at the same time it is efficient and less time consuming. The purpose of this project was to develop a web application and an android application for purchasing items from a shop. This project helped us in gaining valuable information and practical knowledge on several topics like designing web pages using html & css, usage of responsive templates, designing of android applications, and management of database using mysql . The entire system is secured.

Also the project helped us understanding about the development phases of a project and software development life cycle. We learned how to test different features of a project. This project has given us great satisfaction in having designed an application which can be implemented to any nearby shops or branded shops selling various kinds of products by simple modifications. There is a scope for further development in our project to a great extend.

A number of features can be added to this system in future like providing moderator more control over products so that each moderator can maintain their own products. Another feature we wished to implement was providing classes for customers so that different offers can be given to each class. System may keep track of history of purchases of each customer and provide suggestions based on their history. These features could have implemented unless the time did not limited us.

At the end it is concluded that we have made effort on the following points:

1. A description of the background and context of the project and its relation to work already done in the area.
2. Made statement of the aims and objectives of the project.
3. The description of purpose, scope and applicability.
4. We define the problem on which we are working in the project.

Future

6.2 Future Enhancement

We think that not a single project is ever considered as complete forever because our mind is always thinking new and our necessities are also growing. Our application is also, if you see at first glance that you find it to be complete but we want to make it still mature and fully automatic. As system is flexible you can generate more report and screen as and when required, the system is modified in future as per the owner requirement. We have left all the options open so that if there is any other future requirement in the system by the user for enhancement of the system then it is possible to implement them in the last we would like to thank all the people involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is developed there by underlying success of process.

Limitations

6.3 Limitations of the project on Quick Grocery Store:

Although I have put my best efforts to make the software flexible, easy to operate but limitations cannot be ruled out even by me. Though the software presents a broad range of options to its users some intricate options could not be covered into it.

Partly because of the logistic and partly due to the lack of sophistication. Paucity of time was also major constraint, thus it was not possible to make the software foolproof and dynamic. Lack of time also compelled me to ignore some part such as storing the old result of the candidate etc.

Considerable efforts have made the software easy to operate even for the people not related to the field of computers but it is acknowledged that a layman may find it a bit problematic at the first instance. The user is provided help at each step for his convenience in working with the software.

Bibliography

To develop this project we used the knowledge of “C# Language” as Front-End and “Sql Server 2008 R2” as Back-End from the books mentioned below:

1. Microsoft SQL Server 2008 R2 Database Design & Implementation
2. C# and ASP.NET
3. Complete CSS Guide, Maxine Sherrin and John Allsopp-O'Reilly Media; September 2012

Websites:

http://en.wikipedia.org/wiki/Online_shopping/
<http://www.w3schools.com>
<http://www.google.com>
<http://www.msdn.com>
<http://slideshare.org>

Thank
you!